QoE-driven Dynamic Adaptive Video Streaming Strategy with Future Information

Li YU, *Student Member, IEEE,* Tammam TILLO, *Senior Member, IEEE,* and Jimin XIAO, *Member, IEEE*

***Abstract*—Dynamic Adaptive Video Streaming over HTTP (DASH) has become the de-facto video delivery mechanism nowadays, which takes advantage of the existing low cost and wide-spread HTTP platforms. Standards like MPEG-DASH defines the bitstreams conformance and decoding process, while leaving the bitrate adaptive algorithm open for research. So far, most DASH researches focus on the CBR (constant bitrate) video delivery. In this paper, VBR (various bitrate) video delivery is investigated in the on-demand streaming scenario. Detailed instant bitrates of future segments are exploited in the proposed adaptation method to grasp the fluctuation traits of the VBR video. Meanwhile, the adaptation problem is formulated as an optimization process with the proposed internal QoE goal function, which keeps a good balance between various requirements. Besides, the parameters within the internal QoE function can be tuned to guarantee the flexibility of meeting different preferences. The experimental results demonstrate that our proposed QoE-based video adaptation method outperforms the state-of-the-art method with a good margin.**

***Index Terms*—DASH, variable bitrate streaming , QoE, on-demand video streaming.**

I. Introduction

It is indicated by Cisco that 64% of the Internet traffic were made up of videos in 2014, and will be 80% in 2019 [1]. It is difficult for the traditional RTP (Real-time Transport Protocol) video streaming based methods [2] to meet this challenge. This is because RTP video streaming based methods do not provide good interoperability between different servers and devices. Besides, RTP packets are easily blocked by firewalls. Also, RTP video streaming requires a lot of resources to maintain separate streaming sessions for each server-client pair. Thus, based on the widely deployed HTTP (HyperText Transfer Protocol) networks, MPEG Dynamic Adaptive Streaming over HTTP (DASH) [3] was developed and standardized, which overcomes various drawbacks of RTP video streaming.

MPEG-DASH enables the adaptivity to the fluctuations of network throughput and capabilities of client devices. This adaptivity is enabled by preparing representations of various qualities for each video [4], along with associated metadata describing the characteristics of these different representations [5]. Meanwhile, one video is divided into a sequence of segments in time domain. These segments provide the feasibility to adapt the video quality to the network bandwidth with low latency. Based on the metadata and network condition, the client sends requests to the server to download proper representations. The mechanism of choosing a proper representation to download is an important research topic for DASH, which is also the target of this paper. To sum up, DASH appeals to the market because of the following reasons: firstly, it takes good advantage of content delivery networks (CDN), which is widely deployed in today's Internet. Secondly, it is based on HTTP protocol, which is firewall friendly. Last but not the least, it transfers the management of streaming from server side to client side, which saves much server resources and allows for dynamic flexibility.

This work was supported by the National Natural Science Foundation of China (NO. 61210006, NO.61501379 and NO.61501074) and the Jiangsu Science and Technology Programme (BK20150375). (Corresponding author: Jimin XIAO.)

Li Yu, Tammam Tillo and Jimin Xiao are with the Department of Electrical and Electronic Engineering, Xian Jiaotong-Liverpool University, 111 Ren Ai Road, Suzhou, P.R. China, E-mail: (li.yu12@xjtlu.edu.cn, tammam.tillo@xjtlu.edu.cn, jimin.xiao@xjtlu.edu.cn).

There are two approaches to generate different representations of one video, namely CBR (constant bitrate) mode and VBR (variable bitrate) mode. The bitrate is almost constant across the whole video for the CBR mode. While for the VBR mode, the bitrate varies according to the contents of the video. VBR mode is commonly used in many video coding scenarios, such as using coding standards like MPEG-2, MPEG-4 Part 10/H.264 [6]. This is because VBR mode strives to maximize the global quality of the encoded media by allowing a higher bitrate to be allocated to the more complex segments of media files while less is allocated to less complex segments [7]. For example, using HEVC (High Efficiency Video Coding) [8] or H.264/AVC in VBR mode allows to guarantee a constant quality level across different frames, thus minimizing quality fluctuation and the associated visual discomfort. Consequently, the bitrate of each frame varies according to the complexity of the content.

Given the fact that the bitrate fluctuates a lot in the VBR video, it is of significant importance to explore this characteristic in a bitrate adaptation algorithm. However, this information is not specified in the metadata in MEPG-DASH standard. Instead, a general bitrate value of a bunch of frames (defined as representation in DASH) is conveyed to the client. Thus, the adaptation algorithm at client side can only use this general information, which does not contain the detailed fluctuation characteristic. As a result, the mismatch between accurate instant bitrate and general bitrate leads to non-optimal decisions. To tackle this mismatch problem, [9] proposed to include the instant bitrate information in the metadata. However, this proposal did not provide a solution on how to use this information. Besides, there are also some works that attempt to estimate the instant bitrate [10]–[12], so as to restore this information at the client side. However, the estimation precision is limited. In our work, the accurate instant bitrate information will be included at the server side in the extension part of the metadata and sent to the client.

In this work, a QoE-based video bitrate adaptation method is proposed in the scenario of VBR coding mode and on-demand streaming [13]. The usage of accurate instant bitrate of each segment is one of the main contributions of this work. Other contributions are listed as follows. Firstly, the adaptation problem is modeled as an optimization problem, which tries to maximize the quality of experience (QoE) for the whole sequence. Secondly, the optimization problem of the whole sequence is solved by breaking it into sub-optimization problems of each segments to meet the real-time constrains. The goal function of each sub-optimization problem is formulated as "Internal QoE", which explicitly accommodates the need of a sustainable buffer reservation for future streaming. The overall QoE is optimized by combining all the sub-optimization solutions. Thirdly, the weights in the internal QoE metric can be flexibly tuned to meet different requirements. High preference of certain aspect can be achieved by assigning a high weight for the corresponding factor, which allows to tune the streaming session to match the needs of different clients. As demonstrated in the experiments, the proposed method performs better than two typical heuristic methods in VBR modes, with over 27%, 138% gains in smooth network and 78%, 172% gains in fluctuated network respectively.

The paper is organized as follows. In Section II, The DASH standard, as well as related works will be introduced. Next, the problem framework and benchmark methods will be described in Section III. While in Section IV, the proposed method is stated in detail. After that, experiments and discussions are presented in Section V. Finally, conclusions are provided in Section VI.

II. Overview of DASH and Related Works

A. Overview of DASH Standard

As shown in Fig. 1, a typical DASH system consists of a HTTP server and a DASH client. They communicate with each other through the HTTP network. In the HTTP server, video contents of different versions and their description files are stored. Different versions share the same video content, but are encoded using different settings, like resolution, frame rate, QP (Quantization Parameter) and so on. These different versions are called representations in DASH and they provide multiple choices for adaptation. All the representations form an adaptation set. While audio and subtitles form other adaptation sets. For each video representation, it is divided in time domain into several chunks. The chunks are named as segments in the DASH standard. Each segment usually lasts for 2 seconds long [14]–[16]. Each segment is stored as an independent file, which is associated with an URL (Universal Resource Locator) address. In the corresponding description file, the URL addresses and other characteristics, like bandwidth, resolution, media types and program timing are recorded. This description file is called Media Presentation Description (MPD) in DASH, and it is an XML (Extensible Markup Language) document. It describes a hierarchical manifest of the available content and its various versions.

As for the DASH client, it will first obtains the MPD file. After parsing the MPD file, the client decides which segment to request based on the parsed information and network condition. The client sends HTTP GET request to fetch the segment. After accumulating enough buffer reservation, the client starts to play. Meanwhile, following segments will be obtained based on the MPD file, as well as monitored network bandwidth trend to avoid buffer underflows. The intelligence behind the decisions lies in the control heuristics module, which usually tries to provide better video quality while maintaining adequate buffer reservation for continuous playout. This is not defined in the DASH standard, and it is open for research. There are already many related works on this topic, which will be summarized in the following part.

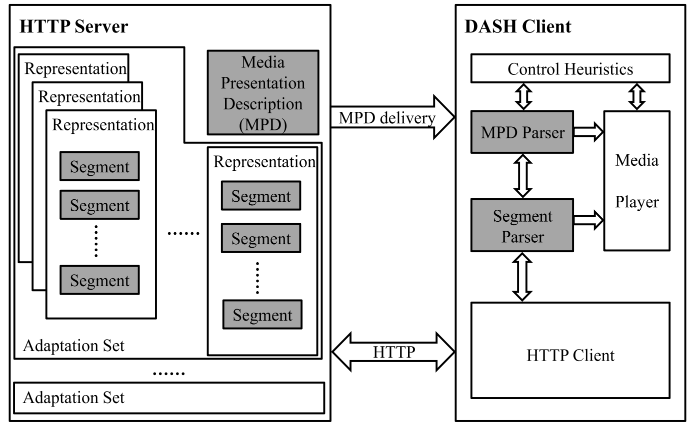


Fig. 1. Scope of the MPEG-DASH standard. The shadowed blocks are defined in the standard, while others are open for development.

B. Related Works

In this part, several classic adaptation methods are described. The existing adaptation methods can be roughly classified into two categories, namely the heuristic rules-based methods and model-based methods. For the heuristic rules-based methods, they set up fixed strategies for different cases. As for the model-based methods, they treat the adaptation problem using existing models or transfer it into an optimization problem. These methods use both the network bandwidth information and the buffer reservation information.

The heuristic rules-based methods can be further classified into two subcategories based on the knowledge used to obtain the strategies. One subcategory is throughput-based methods, which only use the network bandwidth as reference [17] [18]. As the throughput is used to make the decision for future segment, it needs to be estimated. The simplest way of estimation is to use the throughput of previous time slot, which can be calculated as the ratio of data size and the delivery duration of previous segment. However, this method suffers from short-term fluctuations. Thus, a smoothed throughput measurement method is proposed in [17]. This paper computes the throughput as the average download rates of the previous N seconds and tries to keep the requested bitrate around the throughput. With the smoothed bandwidth, the adaptation will be more stable and incurs less quality switches. Another algorithm which uses smoothed HTTP throughput measurement is [18]. Based on the estimated throughput, it proposes a conservative step-wise up switching and aggressive down switching mechanism of representations. This method guarantees a timely adaptation to throughput, as well as reduced buffer overflow and underflow. The other subcategory is buffer-based method [15] [16], which additionally uses the length of buffer reservation information. A partial-linear trend prediction model is proposed in [16] to accurately estimate the trend of client buffer level variation. Based on the estimation, the smoothness in the rate adaptation process is improved. While in [15], the future length of buffer reservation is estimated based on a trellis representation. The results shows that a smooth video quality is provided with buffer underflows eliminated. However, there is a main problem of the heuristic rules-based methods that they are deterministically tailored to specific network configurations.

When it comes to the model-based methods, more flexible solutions are provided comparing to the heuristic rules-based methods. The rate adaptation behaviors are flexibly adapted to the dynamic settings of the network. [19] utilizes the reinforcement learning method to infer the optimal decisions trained in a simulated network. The action is the request of segment with certain bitrate, while the reward is the QoE estimation in the reinforcement learning method. With the QoE as the reward, human perception factor is directly involved in the algorithm. The reinforcement learning method is also introduced in [20] and [21] with the proposition of Q-Learning based clients. These two works can dynamically adjust the streaming behavior according to the current network status while maximizing the QoE. In [22], a subjective study to identify the impact of adaptation parameters on QoE is conducted. Based on this study, it proposes a method to compute the QoE-optimal adaptation strategy for DASH with mixed-integer linear programming. Similarly, [23] proposes a QoE-aware DASH system (QDASH). Besides, it proposes a probing-based network measurement method to facilitate the video quality selection. In [24], Markov Decision Process (MDP) is used to handle the stochastic decision problem, which minimizes both the number of starvation and the number of quality level changes and maximizes the quality level. The overhead of MDP based DASH approaches is analyzed and reduced in [25]. The work in [26] uses stochastic dynamic programing (SDP) techniques to achieve the tradeoff between requested quality and resulting video freezes. It considers two aspects to make the decision. One is that the requested average bitrate should be close to or below the measured bandwidth. Another is that the length of buffer reservation should be around a predefined target value. In general, most of these works are based on the CBR-mode videos. However, VBR-mode videos are also common and easy to produce. In addition, VBR-mode videos guarantee higher global quality than CBR-mode videos.

Thus, our paper will investigate the VBR-mode videos. The work [15] mentioned before also works on the VBR-mode videos, which is one of the benchmarks in our experiment. In that work, estimated bitrates of following segments are used to assist the decision, which may not be accurate. Thus, the accurate instant bitrate of each segment will be used in our proposed decision procedure. The accurate bitrate information will be sent along within the extension part of MDP file, which is standard compliant. Such modification to MDP file is also proposed in [9]. Based on this information, the mismatch between instant bitrate and specific bitrate is avoided. Thus, the decision is more accurate. Besides, the adaptation method is transformed into an optimization problem, which tries to maximize the overall QoE.

TABLE I
DESCRIPTIONS OF KEY SYMBOLS

Symbol	Definition
L	total number of available bandwidth state
M	total number of available quality level for the video
N	total number of segments in one video
$B_i(1 \leq i \leq L)$	all available bandwidth state
$P_i(1 \leq i \leq L)$	probability of each available bandwidth state
$P_{i,j}(1 \leq i, j \leq L)$	transition probability from bandwidth state B_i to B_j
$Q_i(1 \leq i \leq M)$	all available quality level, which $Q_i = i$
τ	duration of each segment
$t_i(1 \leq i \leq N)$	index of each decision point
$b_i(1 \leq i \leq N)$	index of bandwidth level at decision point t_i, $1 \leq b_i \leq L$
$b_i^{'}(1 \leq i \leq N)$	index of estimated bandwidth level at decision point t_i, $1 \leq b_i^{'} \leq L$
$q_i(1 \leq i \leq N)$	index of requested quality level at decision point t_i, $1 \leq q_i \leq M$
$r_{i,q_i}(1 \leq i \leq N, 1 \leq j \leq M)$	bitrate of segment with quality level j at decision point t_i, $r_{i,q_i} \in \mathbb{R}$
Θ	a chain of bandwidth levels chronologically, i.e.$\Theta = \{b_1, b_2, ..., b_N\}$
Ψ	a chain of requested quality levels, i.e.$\Psi = \{q_1, q_2, , q_N\}$
$T_i, 1 \leq i \leq N$	length of buffer reservation in time domain at decision point t_i
$T_i(\Theta, \Psi), 1 \leq i \leq N$	estimated length of buffer reservation in time domain at decision point t_i
$T_i^s, 1 \leq i \leq N$	duration of starvation at decision point t_i
$T^s(\Theta, \Psi)$	total starvation time for one sequence
$T^t(\Theta, \Psi)$	total playout time for one sequence, including the starvation durations
T^b	Size of buffer at the client side
l	the number of future segments involved in the decision for the current segment
Th	the safety threshold of buffer reservation that guarantees a smooth playout (in general benchmark)
$[T^{max}, T^{min}]$	constraint buffer range (in future benchmark)
λ	the weight of buffer reservation change factor in the internal QoE metric

III. Adaptation Problem Formulation

In this section, related concepts of the adaptation algorithm are described, including Markov channel model and Quality of Experience (QoE). Besides, the two benchmark methods are introduced. Important notations and corresponding definitions are listed in Table I.

A. Markov Channel Model

The wireless channel is modeled using finite-state Markov model and first-order Markovian assumption [27]. Assume there are L states of bandwidth level, namely $\{B_1, B_2, \ldots, B_L\}$. The probabilities of each bandwidth levels are $\{P_1, P_2, \ldots, P_L\}$. As it is based on the first-order Markovian assumption, the current bandwidth level is statistically independent of all other past and future bandwidth levels, except the previous bandwidth level. Thus, the transition probability is between two bandwidth levels, which are consecutive in time. The transition probability from B_i to B_j is defined as $P_{i,j}$. Then, the transition matrix is as follows:

$$A = \begin{bmatrix} P_{1,1} & P_{1,2} & \cdots & P_{1,L} \\ P_{2,1} & P_{2,2} & \cdots & P_{2,L} \\ \cdots & \cdots & \cdots & \cdots \\ P_{L,1} & P_{L,2} & \cdots & P_{L,L} \end{bmatrix} \tag{1}$$

In this paper, a five-state Markov model is employed. The probability of each state is deduced from the transition matrix, which represents a stable probability distribution for each state in the current network. This helps to reduce the influence of initial bandwidth state settings. As for the transition matrix, one bandwidth level will not jump to a non-adjacent level, that is,

$$P_{i,j} = 0, \text{if } |i-j| > 1. \tag{2}$$

Thus, the bandwidth level only jumps to the neighboring higher or lower bandwidth level, or stays in the current level.

B. Quality of Experience

The Quality of Experience (QoE) is a concept of subjectively perceived quality, which takes into account how consumers perceive the overall quality of a service [28]. Thus, QoE is regarded as the goal of our proposed adaptation algorithm. As indicated in [28]–[31], QoE is mainly influenced by three key factors, namely requested media quality, quality switching frequency and starvation events. Although startup delay (the period from time starting-to-download to time starting-to-play) is also an important aspect, a fixed startup delay (10s) is set in this paper. Thus, it is not incorporated in the QoE evaluation as in [32].

Assume there are totally N segments in one video sequence. Each segment lasts for τ seconds. The DASH client requests segments of proper quality level according to the available bandwidth. The requested quality levels are $\{q_1, q_2, \ldots, q_N\}$ correspondingly, which is denoted as the requested media sequence Ψ. While the bandwidth for downloading each segments are $\{b_1, b_2, \ldots, b_N\}$, which is labeled as bandwidth chain Θ. Then, the average requested media quality $E(\Psi)$ can be denoted as the average of all requested quality levels:

$$E(\Psi) = \frac{1}{N}\sum_{i=1}^{N} q_i. \tag{3}$$

The quality switching frequency $V(\Psi)$ can be evaluated as the average times of quality change between neighboring segments.

$$V(\Psi) = \frac{1}{N-1}\sum_{i=1}^{N-1} |q_{i+1} - q_i|. \tag{4}$$

While the starvation events under bandwidth chain Θ can be measured as the ratio of starvation event in time domain, i.e. total starvation time $T^s(\Theta, \Psi)$ over the total displaying time $T^t(\Theta, \Psi)$ as shown in the following equation:

$$P^s(\Theta, \Psi) = \frac{T^s(\Theta, \Psi)}{T^t(\Theta, \Psi)}; \tag{5}$$

where

$$T^t(\Theta, \Psi) = N * \tau + T^s(\Theta, \Psi). \tag{6}$$

Starvation event happens when the buffer becomes empty. Assume q_i is requested for i^{th} segment, and its corresponding bitrate is r_{i,q_i}. The network bandwidth is b_i at the downloading period, while the length of buffer reservation is T_{i-1} before downloading. Then, the corresponding starvation duration can be calculated as follows:

$$T_i^s = max(\frac{r_{i,q_i} * \tau}{b_i} - T_{i-1}, 0). \tag{7}$$

Thus, if enough buffer reservation is maintained before loading, i.e. $T_{i-1} \geq r_{i,q_i} * \tau / b_i$, the starvation will not happen. Otherwise, the starvation duration is the difference between download time and length of buffer reservation. Then, the total starvation time can be calculated as the sum of all starvation durations:

$$T^s(\Theta, \Psi) = \sum_{i=1}^{N} T_i^s. \tag{8}$$

Up to now, all these three factors are defined. It is worth noticing that all of them are normalized, so as to guarantee fair comparisons between video sequences of different lengths. However, the goal of these three factors are conflicting with each other. When the goal is to minimize the starvation events, smallest available bitrates will always be selected. As a result, a low average media quality is incurred. Conversely, selecting highest available bitrates may lead to high probability of starvation. Moreover, when the solution tries to have higher media quality under the constraint of low probability of starvation, the quality switching event will inevitably increase. Thus, these three factors are balanced with different weights in the QoE metric, which is calculated as follows:

$$QoE(\Theta, \Psi) = E(\Psi) - w_1 V(\Psi) - w_2 P^s(\Theta, \Psi); \tag{9}$$

where $\{w_1, w_2\}$ are the relative weights with respect to $E(\Psi)$. The weights will be tuned according to the different requirements of the client. The setting of w_2 is motivated by work in [32]. Based on the subjective tests, 10% of starvation ratio is equivalent to 2 levels drop in the quality level. Thus, w_2 is set as 20. While for the setting of w_1, a sensible range is defined based on the following heuristic analysis. The highest available quality level is denoted as Q_M, while the lowest available quality level is denoted as Q_1. Three “extreme” cases of requested quality levels are shown in Fig. 2. Case 1 represents the QoE score corresponding to the maximum fluctuation of quality levels, while case 2 and case 3 represent the ones with highest and lowest average quality respectively. The starvation ratio is assumed to be a constant for all cases. For simplicity, $P^s(\Theta, \Psi) = 0$. Then, the QoE values of these three cases are computed as follows:

$$QoE_{Case1} = \frac{1}{2} * (Q_M + Q_1) - w_1 * (Q_M - Q_1); \tag{10}$$

$$QoE_{Case2} = Q_M; \tag{11}$$

$$QoE_{Case3} = Q_1. \tag{12}$$

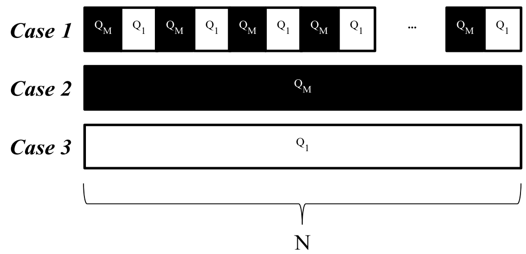


Fig. 2. Three cases of "extreme" requested quality level sequence.

To most of the audience, QoE_{Case1} should be between QoE_{Case2} and QoE_{Case3}, that is $QoE_{Case3} \leq QoE_{Case1} \leq QoE_{Case2}$. Then, the range of w_1 is $[-\frac{1}{2}, \frac{1}{2}]$. It is worth to point out that w_1 should be a positive value since large quality fluctuation is regarded as a negative influence on QoE. Thus the range of w_1 is as follows:

$$0 \leq w_1 \leq \frac{1}{2}. \tag{13}$$

The weights can be flexibly tuned within a reasonable range according to different preferences. For example, if the client is not sensitive to quality level switching, a lower w_1 can be set to give a higher priority to the other two factors. If the client prefers high quality than fluent playout, then w_2 can be lowered to concentrate more on the quality factors.

C. Benchmark Methods

In this section, a general framework of the bitrate adaptation problem, as well as two benchmark adaptation strategies, namely general buffer-based method and future buffer based method, will be presented respectively.

1) Framework: The adaptation strategy is applied sequentially to consecutive decision points $\{t_1, t_2, \ldots, t_N\}$. At one decision point t_i, one quality level $q_i, (1 \leq i \leq N)$ will be selected among all available quality levels $\{Q_1, Q_2, \ldots, Q_M\}$ based on the buffer status and the predicted bandwidth $b_i^{'}$. The bandwidth prediction methods used in both benchmarks are the same. A simple aggressive method [10] is employed, where the throughput of downloading previous segment is used as the prediction of current bandwidth. It is shown in [10] that the aggressive method obtains satisfactory result similar to the proposed method in [10], when the duration of segment is short (e.g. 2s or 4s). During the downloading time of one segment, the bandwidth $b_i, (1 \leq i \leq N)$ is assumed to be stable. Besides the quality level, each segment is also associated with its bitrate value $r_{i,q_i}, (1 \leq i \leq N)$. At start, the segment with the lowest quality level will be chosen (i.e. $q_1 = Q_1$). Then, based on the adaptation strategy, the following segments will be requested, downloaded and stored in the buffer. The length of buffer reservation at each decision point t_i is $T_i (1 \leq i \leq N)$. When the filled buffer reservation is more than $10s$ (i.e. $T_i = 10, i = 5$), the client will start to play. Then, the buffer is influenced by both downloading speed and playout speed as shown below.

$$T_{i+1} = max(T_i - \frac{r_{i,q_i} * \tau}{b_i} + \tau, 0). \tag{14}$$

Buffer reservation T_i should be limited under the buffer size T^b at the client side. The download process stops, when buffer overflows, i.e. $T_i \geq T^b$.

2) General Buffer-based Method: This benchmark method is developed based on several related works [17] [18] [33], which represents the core idea of general buffer-based methods. The adaptation decision is made based on the predicted bandwidth and length of buffer reservation. At first, the estimated bandwidth $b_i^{'}$ is compared to bitrates of segments at all available quality levels r_{i,q_i}. Then, based on whether the length of buffer reservation T_i reaches the safety threshold Th, one quality level up or down is selected.

$$q_i = \begin{cases} Q_k \quad , r_{i,k} \leq b_i < r_{i,k+1} \text{ and } T_i < Th; \\ Q_{k+1}, r_{i,k} < b_i \leq r_{i,k+1} \text{ and } T_i \geq Th. \end{cases} \tag{15}$$

If $T_i < Th$, a lower quality is chosen, and vice versa. Instant bitrates of future segments can also be used in this method. Instead of using bitrate of current segment r_{i,q_i}, average bitrate of current and future l segments $\sum_i^{i+l} r_{i,q_i}/(l+1)$ is used to compare with the estimated bandwidth $b_i^{'}$ in the first step.

3) Future Buffer based Method: Similar to our proposed method, the work [15] also employs future instant bitrate information to assist the adaptation in VBR scenario. Thus, it is used as another benchmark for comparison. The instant bitrates of future segments used in [15] are predicted from the downloaded segments following the prediction mechanism in [10]. For fair comparison, accurate instant bitrates will be used in this algorithm. Based on the instant bitrates of future l segments, as well as the predicted bandwidth $b_i^{'}$, this work builds a trellis representation to estimate future buffer level $T_i^{'}$ following the rules of Up-case and Down-case. The goal of this method is to keep the buffer T_i within a given range $[T^{min}, T^{max}]$. The trellis representation contains a path of quality request decisions for future l segments. Each time one segment is downloaded, the buffer status would be checked and compared with the estimated buffer level. Once the difference is larger than τ, a new trellis representation would be built to replace the old one. Therefore, the path is only updated at some of the decision points, i.e. the adaptation algorithm is not always performed before downloading each segment. As a result, the adaptivity is limited. Whereas in our proposed method, there is no such limitation. Because the adaptation algorithm is performed for all the segments. Thus, better performance is achieved using the proposed method. Detailed discussion will be presented in the experimental section.

IV. Proposed Method

A. Overview of the Proposed Method

The motivations of the proposed method will be explained in this section. In general, the proposed method tries to optimize QoE by exploring the future information (instant bitrates of segments to download) with a probabilistic bandwidth prediction model. To sum up, the motivations are two folds:

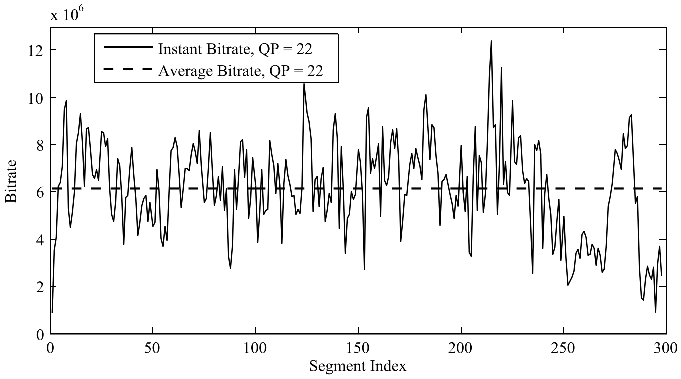


Fig. 3. The bitrates versus segment indexes of sequence basketballPass are plotted when QP = 22. Average bitrate of the whole sequence is shown in dashed line for comparison. Similar phenomena happens for other QP settings and other video sequences.

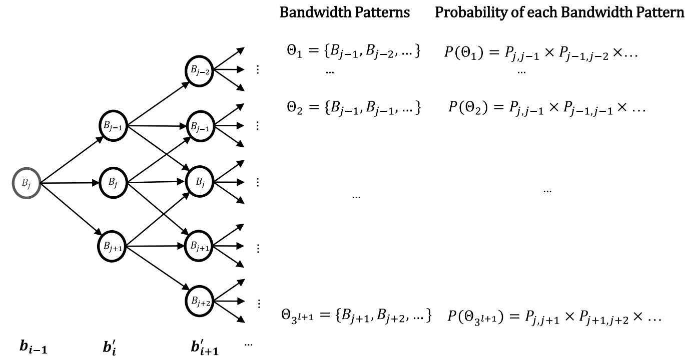


Fig. 4. All possible bandwidth patterns over current and future l time slots $[t_i, t_{i+l}]$. $b_{i-1} = B_j$ is the bandwidth for downloading previous segment.

1) QoE Optimization: The viewing experience of the end user is regarded as the evaluation criteria of adaptation performance. There are two major ways to evaluate the viewing experience of users, including subjective way like MOS and objective way like QoE. The former one is usually time-consuming. Thus in this paper, the objective QoE metric is adopted as in other DASH works [27], [32]. Then, the adaptation problem is transformed into the optimization of QoE. However, the QoE metric is an overall evaluation of the adaptation results, which can only be obtained when all the segments are downloaded. Thus, internal QoE (QoE^{inter}) is proposed as the medium-term goal, which can be evaluated for each segment. By doing so, the global optimization problem is divided into a collection of simple and real-time sub-optimization problems. The details of the internal QoE metric will be discussed later.

2) Future Information: The future information refers to the instant bitrate of future segments. It is involved in each sub-optimization problem to provide insights into the future, which will lead to more globally optimal results than just investigating current and previous information. The reason to use future information is that bitrates of future segments differ from that of current segment, as well as average bitrate. This is particularly important for VBR videos, where the instant bitrates of segments uctuates a lot, as could be seen in the Fig. 3. Furthermore, only a small portion of the instant bitrate is similar to the average bitrate. It is worth highlighting that this is a common phenomenon in most VBR video sequences. As a result, the methods that use the average bitrate in the adaptation mechanism will incur huge mismatches in the VBR mode videos, which will lower the overall performance. Thus, to avoid this effect, actual instant bitrates of future segment are used in the proposed method by inserting them in the extension part of the MPD file.

Moreover, the decision at the current segment will influence the buffer status, which will further influence future decisions. For example, if the highest quality level is chosen for the current segment and the downloading time is higher than the duration of this segment. Then, a reduction in length of buffer reservation is caused. With lower length of buffer reservation, future decisions will prefer to request lower quality levels to fill the buffer. Thus, it is better to consider the future trend to achieve global optimization. In the the proposed method, future l segments after the current segment will be investigated to help the decision for the current segment.

B. Markov Channel Model

As the proposed method exploits l future segments for bitrate adaptation, the bandwidth estimation is needed. In this work, the smoothed throughput is used for bandwidth estimation. Instead of using moving average as throughput estimation method, a heterogeneous Markov model is used to predict the future bandwidth [34]–[37]. The transition matrix used here is the same as Equ. (1). Supposing $(i-1)^{th}$ segment has been downloaded under the bandwidth $b_{i-1} = B_j$. The bandwidth for downloading current and future l segments are estimated as $b_i^{'}, b_{i+1}^{'}, ..., b_{i+l}^{'}$, as shown in Fig. 4. As defined in Equ. (1), each state would only jump to neighboring states or stay in the current state. Thus, given $b_{i-1} = B_j$, $b_i^{'}$ could be B_{j-1}, B_j or B_{j+1}. Following this rule, there are totally 3^{l+1} possible throughput chains, as listed in the 'Bandwidth Patterns' column in Fig. 4. The probability of a bandwidth pattern $\Theta_k = \{b_i^{'}, b_{i+1}^{'}, ..., b_{i+l}^{'}\}$ is calculated as

$$P(\Theta_k) = P_{b_{i-1}, b_i'} \times \prod_{j=0}^{l-1} P_{b_{i+j}', b_{i+j+1}'}. \tag{16}$$

Detailed probabilities of each bandwidth pattern in Fig. 4 are shown in the rightmost column. To sum up, the Markov channel model provides all possible bandwidth chains $\{\Theta_1, \Theta_2, ..., \Theta_{3^{l+1}}\}$, along with their probabilities, as the prediction output.

C. Proposed Method in Details

The working flow of the proposed method is shown in Fig. 5. At the beginning, the first segment with the lowest quality is requested to reduce the startup delay. For all the following segments, the proposed adaptation method would decide which quality level to request based on the result of sub-optimization problem. The sub-optimization problem

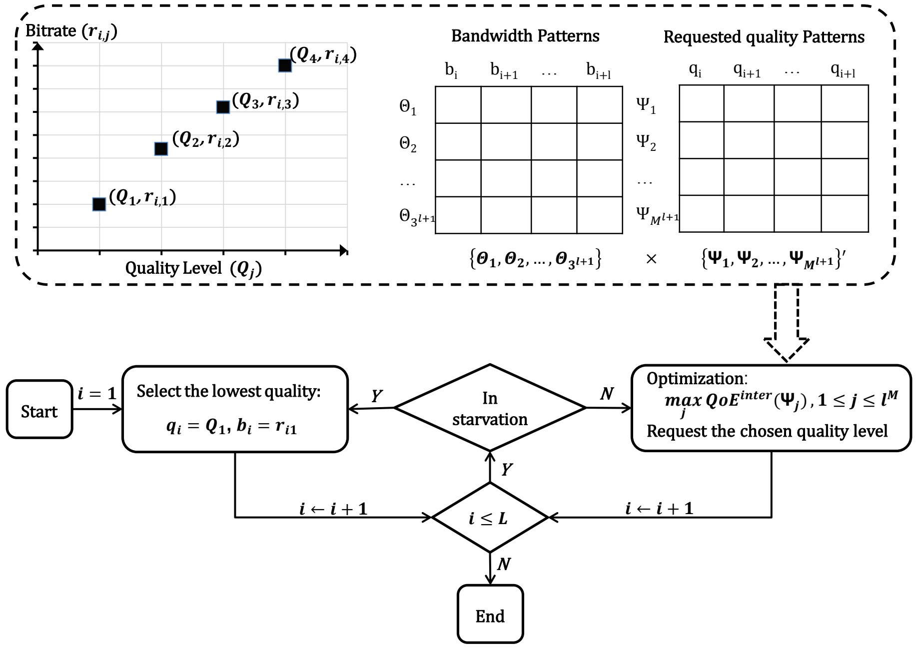


Fig. 5. Flowchart of the proposed method is represented with solid line arrows and boxes. While dashed line arrows and boxes denote the information flow. The streaming process starts with the lowest quality level. Once the buffer is in starvation, the lowest quality level is requested until the starvation ends. When the buffer jumps out of starvation, the decision to choose which quality level follows the result of sub-optimization process. Information needed in the sub-optimization process are shown in the box of dashed line, including the accurate bitrate information, as well as all possible bandwidth patterns and requested quality patterns.

selects the quality level which reaches the maximum expected internal QoE score.

$$\begin{aligned} \max \quad & QoE^{inter}(\Psi_j), \\ s.t. \quad & j \in \{1, 2, \dots, M^{l+1}\}. \end{aligned} \tag{17}$$

The maximization problem is solved by a greedy search approach among all possible requested quality patterns. Each requested quality pattern is a chain of requested quality levels for the current and future l segments. For example, $\Psi_j = \{Q_1, Q_1, \dots, Q_1\}$ is one requested quality pattern, where the quality levels selected for the current and following segments are all Q_1. For each segment, there are M quality levels to be chosen from. Thus, there are M^{l+1} requested quality patterns $\{\Psi_1, \Psi_2, \dots, \Psi_{M^{l+1}}\}$. The expected internal QoE score for each requested quality pattern is calculated as follows:

$$QoE^{inter}(\Psi_j) = \sum_{i=1}^{3^{l+1}} QoE^{inter}(\Theta_i, \Psi_j) * P(\Theta_i), \tag{18}$$

where $QoE^{inter}(\Theta_i, \Psi_j)$ represents the internal QoE score obtained from downloading Ψ_j under bandwidth Θ_i. As described in the previous sub-section, there are totally 3^{l+1} predicted bandwidth patterns with different probabilities. The expected internal QoE score for Ψ_j is an weighted average of internal QoE score under all possible bandwidth patterns Θ_i, with probability $P(\Theta_i)$ as weights.

Finally, the requested quality pattern Ψ_j with maximum expected internal QoE score will be chosen. Then, the quality level of current segment will be selected following this Ψ_j pattern. Since now, the decision has been made, and the request will be sent to the server. Then, the requested quality level as well as the actual network bandwidth will be fed into the next round of decision. If the client is in starvation, the lowest quality level is requested to reduce delay.

In the following section, detailed information of the internal QoE metric, as shown in Equation (20), will be described .

D. Goal Function of Sub-Optimization: Internal QoE Metric

The goal function plays a vital part in the whole optimization process. The internal QoE metric is proposed as the medium-term optimization goal, which leads to a near optimal result of the adaptivity problem. Similar to the QoE metric that has been introduced previously, internal QoE metric also tries to improve the three factors: requested media quality, quality switching frequency and starvation events. The difference lies in that the internal QoE metric evaluates the performance in the middle of the streaming process. In this case, the streaming will need to continue after this evaluation. Thus, it is important to incorporate the future effect into the internal QoE evaluation. Buffer reservation, which is the common fortune across the whole streaming process, plays a key role in future effect. Thus, the change in length of buffer reservation, called "buffer change" for short, caused by the current decision is included in the internal QoE metric. Different from starvation factor in the QoE metric, which accounts for the starvation probability for now, buffer change factor take responsibility for starvation probability afterwards. Like the three factors in QoE metric, buffer change factor is also normalized as a value per segment. Given a bandwidth pattern Θ and a requested quality pattern Ψ over future $l + 1$ segments, the estimated length of buffer reservation at decision point t_{i+l} can be denoted

TABLE II
AVERAGE BITRATES OF DIFFERENT VERSIONS OF TEST VIDEO SEQUENCE "BIG BUCK BUNNY"

Quality Level	QP	Average Bitrates (kbps)
Q_4	22	733.66
Q_3	27	383.29
Q_2	32	183.06
Q_1	37	88.52

as $T_{i+l}(\Theta, \Psi)$. Then, the normalized buffer change can be calculated as follows:

$$\Delta T_i(\Theta, \Psi) = \frac{T_{i+l}(\Theta, \Psi) - T_{i-1}}{l+1}. \tag{19}$$

It is incorporated into the internal QoE metrics as follows:

$$QoE^{inter}(\Theta, \Psi) = E(\Psi) - w_1 V(\Psi) - w_2 P^s(\Theta, \Psi) + \lambda \Delta T_i(\Theta, \Psi), \tag{20}$$

where λ is the weight of buffer change factor that balances its importance against other three factors. When the buffer has accumulated enough reservations, the buffer change factor will not be that important. That is, the increase in buffer will not be as important as other three factors, while the decrease in buffer will also not cause disastrous results. Thus, λ can be assigned with a relatively small value. On the contrary, when the length of buffer reservation is under the safety threshold, it is of crucial importance to ensure an increasing trend in buffer change. In this case, λ should be set with a relatively high value. To sum up, the setting of λ can be represented as a linear function of buffer reservation with a negative slope, i.e. the bigger $T_{i+l}(\Theta, \Psi)$, the smaller λ. It can be represented as follows:

$$\lambda_i = a * T_{i+l}(\Theta, \Psi) + b, a < 0. \tag{21}$$

It is worth noticing that, the length of buffer reservation used here is the one at decision point $i + l$, which is the final status for current decision. In the following experiment section, Equation (21) will be further investigated.

V. EXPERIMENTS

In this section, the experimental settings is introduced first. Then, the investigation of parameters in the proposed method is discussed. Based on these settings, the comparisons between proposed method and benchmarks in both smooth and fluctuated networks are provided. Finally, the robustness of the proposed method to perturbed bandwidth prediction is shown.

A. Experimental Settings

The proposed method will be evaluated in comparison with two benchmarks as described in Section III. The general buffer-based method and the future buffer based method are called "general benchmark" and "future benchmark" respectively for simplicity. Th is set as $10s$ in the general benchmark method. Buffer range $[T^{min}, T^{max}]$ in the future benchmark method is set as $[10s, 30s]$ according to [15]. The interaction between the DASH server and client is simplified and simulated using Matlab. The test video sequence "Big buck Bunny" [38] is encoded by JM 17.0 with the main profile of AVC (Advanced Video Coding) [39]. Different QPs, namely $\{22, 27, 32, 37\}$, are used to represent different VBR video versions. That is, four quality levels will be provided, i.e. $Q_1 = 1, Q_M = 4$. Each video file has a frame rate of $24fps$ and a resolution of 352×288. Segments are generated with fixed duration $\tau = 2s$ and stored as separate files. The total number of segments is 298. Average bitrates of each quality version is shown in Table II.

For the network simulation, five levels of bandwidth state are used, namely $900, 600, 300, 140$ and 50 kbps. The lowest bandwidth state 50 kbps is lower than the lowest average media bitrate, which is a reasonable arrangement. The transition probabilities between different states are represented by the following transition matrix:

$$A = \begin{bmatrix} 0 & 0.05 & 0 & 0 & 0 \\ 0.03 & 0 & 0.03 & 0 & 0 \\ 0 & 0.03 & 0 & 0.02 & 0 \\ 0 & 0 & 0.02 & 0 & 0.03 \\ 0 & 0 & 0 & 0.06 & 0 \end{bmatrix}. \tag{22}$$

The matrix A represents a smooth network with few bandwidth fluctuations. Besides, a fluctuated network is derived with $10 \times A$ as transition matrix. Both settings are used in the experiments to evaluate the effectiveness of the proposed method for different network scenarios. Totally 2000 unique bandwidth chains are prepared for simulation. Each obtained QoE result is averaged over these 2000 trials to obtain statistical stable results.

B. Investigation of Weights Setting

As mentioned before, one advantage of the proposed QoE-based method is the flexibility to tune the weights of different factors so as to appeal to different demands. Thus, different settings of the w_1 and w_2 in QoE metric are investigated in this part to investigate their influences on final performance. Besides, the parameter λ in the internal QoE metric, which has direct influence on optimization result, is also investigated.

1) w_1 and w_2: The values of w_1 and w_2 decide the preference on different factors. Thus, modifying weights in a proper range would help to meet different requirements and preferences of different users.

The proper range has been analyzed in Section III-B. With the total number of available media quality levels set as 4, the range of w_1 is $[-1/2, 1/2]$. Meanwhile, the setting of w_2 can be determined by mapping the QoE loss caused by starvation events to that of quality degradations.

Within the range, different settings of w_1 and w_2 are assessed. As expected, when the weights changes, the range and the meaning of QoE value would change accordingly. Thus, QoE values are incomparable across different settings of weights. Instead, the scores of three factors are used for comparison here, which are shown in Table III. The first row, i.e. $w_1 = 1/3$ and $w_2 = 20$, is used as benchmark for comparison. With higher w_1, quality variation is reduced

TABLE III
QoE PERFORMANCE WITH DIFFERENT SETTING OF w_1 AND w_2 ($l = 1, \lambda = 0.9$).

w_1	w_2	Average Quality	Quality Variation	Starvation Ratio
$\frac{1}{3}$	20	2.2601	0.0281	0.0347
$\frac{1}{2}$	20	1.9966	0.0135	0.0333
$\frac{1}{3}$	1	2.316	0.0857	0.1132

when comparing the first two rows. Similar observations can be found with $w_2 = 1$, where the starvation ratio becomes higher. w_1 and w_2 are fixed as $1/3$ and 20 respectively in the following experiments.

2) λ: As an important parameter in the proposed method, λ balances the tradeoff between change of buffer reservation and other three QoE factors. The setting of λ has a direct influence on the adaptation decision, as well as the final QoE performance. Experiments under different network settings are evaluated. Generally, the peak QoE values are obtain when λ is 0.9. The linear λ is investigated based on the best fixed value for λ, i.e. 0.9. The average length of buffer reservation is around $48s$ in that setting. Thus, a and b would roughly satisfy the following equation $0.9 = a{*}48{+}b, a < 0$. Different combinations of a and b are evaluated. The combination ($a = 1.86, b = -0.02$) is selected and will be used in the following experiments, so

$$\lambda = 1.86 - 0.02 \times T_{i+l}(\Theta, \Psi). \tag{23}$$

The QoE performance improves about 8% with linear λ, comparing to the fixed λ.

C. Comparison to Benchmarks

In this part, the proposed method is compared to the two benchmarks. To demonstrate the performance under different network scenarios, both smooth and fluctuated networks are used in the experiment.

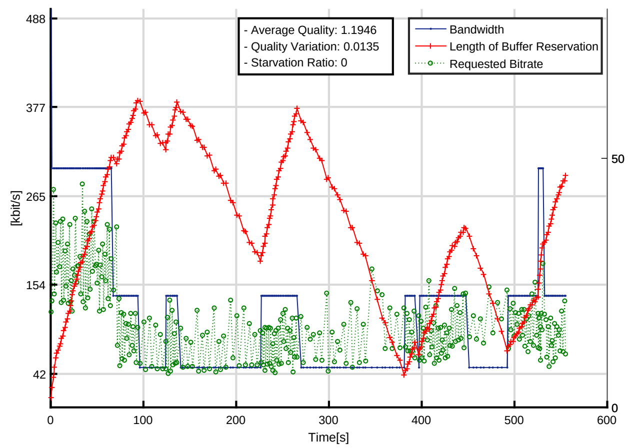
The bandwidth, requested media bitrate and length of buffer reservation over one sample adaptation session are illustrated in Fig.6. The proposed method, the general benchmark and the future benchmark, which uses predicted and actual bitrate, are reported. When the length of buffer reservation falls to 0, it means the starvation happens. It can be found that, there is much less starvation events happening for the QoE-based method than the benchmarks. When the bandwidth is even lower than the lowest media bitrate (bandwidth valley), starvation could still be avoided by taking advantage of the previously accumulated buffer reservations as shown in Fig.6-(a). As for the general benchmark, the requested bitrates closely follow the fluctuation of bandwidth, even during the very short peak at $660s$ in Fig.6-(b). As a result, the buffer reservation is always at a low level, which makes it vulnerable to starvation during bandwidth valleys. While for the future benchmark method, the starvation is less severe than the general benchmark. It only happens when bandwidth valley lasts for over $40s$. It is worth to notice that, the future benchmark using actual bitrate avoids the starvation at $550s$ as in Fig.6-(d). While the one using predicted bitrate in Fig.6-(c) fails, which is due to the unprecise bitrate information.

Besides, the variation of requested bitrates in the proposed method is much lower than the benchmarks, which would guarantee a stable watching quality. The corresponding quality variation of the proposed method is 0.0135. While for the general benchmark, the quality variation is 0.5185, nearly 38 times of the proposed method. As for the future benchmarks, the quality variation is almost 10 times of the proposed method. It is worth to notice that, there is less overshoots of requested bitrates for Fig.6-(d), when comparing with the requested bitrates with Fig.6-(c).

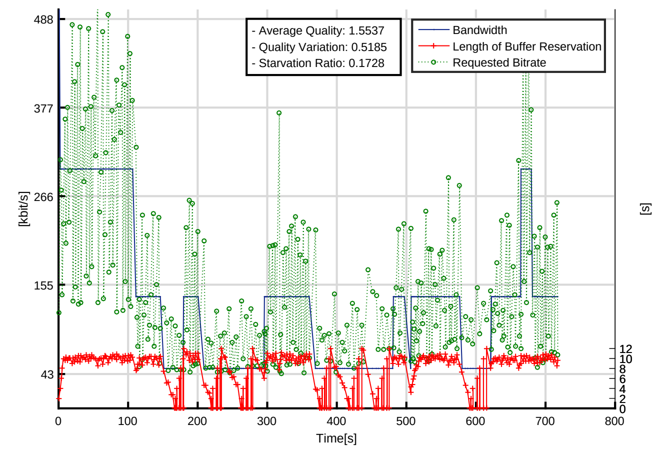
Detailed QoE performances with different look ahead length l are shown in Table IV under both smooth and fluctuated networks. $l = -1$ and 0 represent using the bitrate of previous and current segment respectively, while $l > 0$ denotes that bitrate of future l segments are used. It can be observed that the QoE performance enhances a lot from $l = -1$ to 0 and $l = 0$ to 1 for the general benchmark method. This is rational because more information guarantees wiser decisions. In addition, the increase mainly comes from lower quality variation and starvation ratio. This demonstrates the importance of using future information. When $l \geq 1$, the QoE performance generally remains stable for all methods. This demonstrates that the information of farther segment has less contributions. Based on this observation, l can be set as 1 to obtain a desirable result while maintaining a low computational complexity. When it comes to the future benchmark method, the one using actual bitrate always gets better performance than the one using predicted bitrate. This reveals the importance to use the actual bitrates, if possible, rather than the predicted ones. Generally in the smooth network, the proposed method outperforms the benchmarks, with over 27% and 138% improvement in QoE value comparing to future benchmark method with actual bitrate and general benchmark method respectively. While in the fluctuated network, the improvements are 78% and 172% respectively. To sum up, our proposed method is effective in both smooth and fluctuated networks.

D. Evaluation of Robustness to Perturbed Bandwidth Prediction

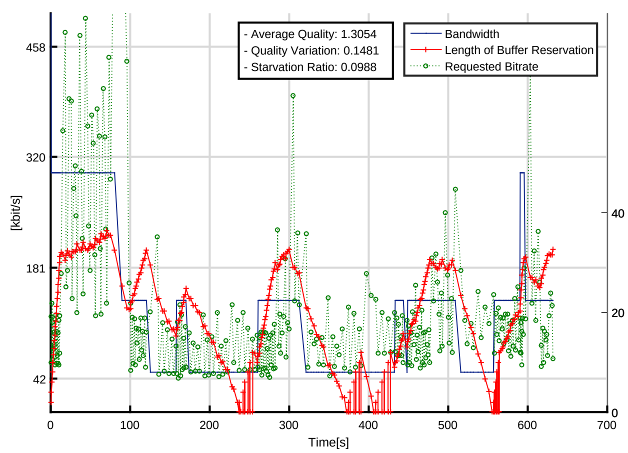
The accuracy of bandwidth prediction has a direct influence on adaptation algorithm. Thus, the robustness of the proposed method to perturbed bandwidth prediction is evaluated in this part. A perturbed transition matrix is used in the prediction process to investigate its influence on the final result. As shown in Table V, the first row, which uses accurate transition matrix for bandwidth prediction, is used as benchmark for comparison. The second row uses a random perturbed transition matrix with similar order of magnitude as accurate one. The QoE performance is 1.67, which is similar to the result of accurate one. While the third row uses a random perturbed transition matrix with higher order of magnitude. The QoE score is 3.6% lower than the result of accurate one. These results demonstrate the robustness of the proposed method to perturbed bandwidth prediction.



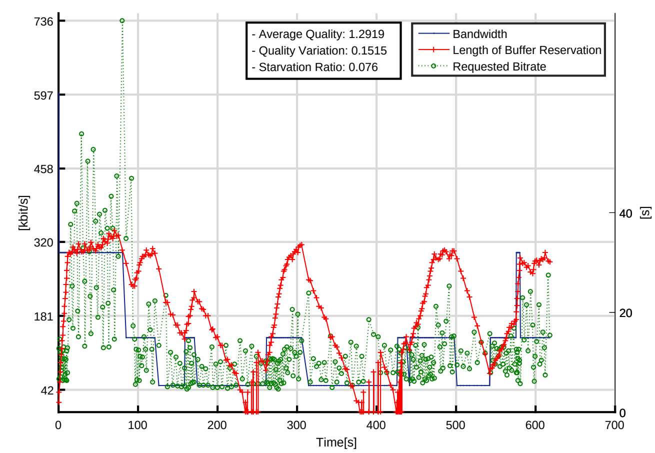
(a) Proposed QoE-based Method



(b) General Benchmark Method



(c) Future Benchmark Method using Predicted Bitrate



(d) Future Benchmark Method using Actual Bitrate

Fig. 6. Illustration of bandwidth, requested media bitrate and length of buffer reservation for both benchmarks and proposed method for $l = 1$. Both future benchmark method using (c) predicted and (d) actual bitrate are assessed. The detailed values of average quality, quality variation and starvation ratio are also tagged. The right vertical axis are scaled with same maximum value for easy comparison of the length of buffer reservation.

TABLE V
QoE PERFORMANCE UNDER PERTURBED BANDWIDTH PREDICTION.

Transition Matrix used for Bandwidth Prediction	QoE	Average Quality	Quality Variation	Starvation Ratio
$\begin{bmatrix} 0 & 0.05 & 0 & 0 & 0 \\ 0.03 & 0 & 0.03 & 0 & 0 \\ 0 & 0.03 & 0 & 0.02 & 0 \\ 0 & 0 & 0.02 & 0 & 0.03 \\ 0 & 0 & 0 & 0.06 & 0 \end{bmatrix}$	1.68	2.3847	0.0349	0.0345
$\begin{bmatrix} 0 & 0.06 & 0 & 0 & 0 \\ 0.02 & 0 & 0.04 & 0 & 0 \\ 0 & 0.05 & 0 & 0.01 & 0 \\ 0 & 0 & 0.04 & 0 & 0.01 \\ 0 & 0 & 0 & 0.05 & 0 \end{bmatrix}$	1.67	2.3837	0.0325	0.0354
$\begin{bmatrix} 0 & 0.6 & 0 & 0 & 0 \\ 0.2 & 0 & 0.4 & 0 & 0 \\ 0 & 0.5 & 0 & 0.2 & 0 \\ 0 & 0 & 0.4 & 0 & 0.1 \\ 0 & 0 & 0 & 0.5 & 0 \end{bmatrix}$	1.62	2.3397	0.0286	0.0357

VI. Conclusions

In this paper, a QoE-based video adaptation method is proposed to adapt VBR video streaming over HTTP. This method incorporates the QoE evaluation metric, which is the goal of the adaptation problem, into the decision mechanism. Besides, the adaptation problem is transformed into an optimization problem, which is divided into a collection of sub-optimization problems to make the algorithm real-time resolvable. Meanwhile, the instant bitrates of each segment are sent in the extension part of the MPD file to precisely follow the bitrate fluctuation of the VBR video. Experimental results showed the importance of using accurate instant bitrate information and looking ahead into future segments. Also, the proposed method outperforms the two benchmarks by 27%, 138% in smooth network and 78%, 172% in fluctuated network respectively.

References

[1] I. Cisco, "Cisco visual networking index: Forecast and methodology, 2014-2019," *CISCO White paper*, 2014.

[2] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "Rtp: A transport protocol for real-time applications," Tech. Rep., 2003.

[3] T. Stockhammer, "Dynamic adaptive streaming over http: standards and design principles," in *Proc. of the 2nd Annu. ACM Conf. on Multimedia Syst.* ACM, 2011, pp. 133–144.

TABLE IV
QoE PERFORMANCE OF BOTH BENCHMARK METHODS AND PROPOSED METHODS WITH DIFFERENT LOOK AHEAD LENGTH l. FOR THE FUTURE BENCHMARK METHOD, BOTH CASES ARE ASSESSED, INCLUDING USING PREDICTED BITRATE AND USING ACTUAL BITRATE IN THE ADAPTATION MODULE. BOTH SMOOTH (A) AND FLUCTUATED ($10 \times A$) NETWORKS ARE EVALUATED.

Method		l	A				10 × A			
			QoE	Average Quality	Quality Variation	Starvation Ratio	QoE	Average Quality	Quality Variation	Starvation Ratio
General Benchmark		-1	0.45	2.5602	0.5463	0.0966	-0.49	2.5205	0.6158	0.1404
		0	0.63	2.6150	0.5275	0.0905	0.37	2.5226	0.5928	0.0976
		1	0.71	2.5884	0.4374	0.0869	1.18	2.4473	0.4941	0.0552
		2	0.70	2.5903	0.4427	0.087	1.15	2.4509	0.4991	0.0569
		3	0.71	2.5908	0.4286	0.0867	1.20	2.4464	0.4912	0.0542
Future Benchmark	Predicted bitrate	1	1.28	2.3923	0.236	0.0867	1.89	2.0909	0.2723	0.0055
		2	1.30	2.356	0.2217	0.049	1.89	2.0795	0.2549	0.0053
		3	1.30	2.3616	0.1779	0.0504	1.92	2.0789	0.2433	0.0039
	Actual bitrate	1	1.31	2.3888	0.2599	0.0498	1.90	2.0931	0.2654	0.0053
		2	1.32	2.3676	0.2146	0.0487	1.90	2.0818	0.2463	0.0049
		3	1.35	2.3683	0.1587	0.0487	1.92	2.0779	0.2313	0.004
QoE-based Method		1	1.68	2.3847	0.0349	0.0345	2.11	2.1924	0.1958	0.0008
		2	1.70	2.3866	0.0373	0.0337	2.10	2.1784	0.2004	0.0007
		3	1.69	2.3826	0.0373	0.0338	2.08	2.1478	0.1632	0.0007

[4] G. Gao, W. Zhang, Y. Wen, Z. Wang, and W. Zhu, "Towards cost-efficient video transcoding in media cloud: Insights learned from user viewing patterns," *IEEE Transactions on Multimedia*, vol. 17, no. 8, pp. 1286–1296, 2015.

[5] T. C. Thang, Q.-D. Ho, J. W. Kang, and A. T. Pham, "Adaptive streaming of audiovisual content using mpeg dash," *IEEE Trans. Consum. Electron.*, vol. 58, no. 1, pp. 78–85, 2012.

[6] Wikipedia, "Variable bitrate," http://en.wikipedia.org/wiki/Variable_bitrate.

[7] MSDN, "Variable bit rate (vbr) encoding," http://msdn.microsoft.com/en-us/library/windows/desktop/dd743964(v=vs.85).aspx.

[8] G. Sullivan, J. Ohm, W.-J. Han, and T. Wiegand, "Overview of the high efficiency video coding (hevc) standard," *IEEE Trans. Circuits Syst. Video Technol*, vol. 22, no. 12, pp. 1649–1668, Dec 2012.

[9] T. Thang, J. Lee, J. Kang, S. Bae, S. Jung, and S. Park, "Proposal on signaling for dash," *ISO/IEC JTC1/SC29/WG11 m18445, Guangzhou*, 2010.

[10] T. C. Thang, H. T. Le, H. X. Nguyen, A. T. Pham, J. W. Kang, and Y. M. Ro, "Adaptive video streaming over http with dynamic resource estimation," *J. of Commun. and Networks*, vol. 15, no. 6, pp. 635–644, 2013.

[11] A. Bhattacharya, A. G. Parlos, and A. F. Atiya, "Prediction of mpeg-coded video source traffic using recurrent neural networks," *IEEE Trans. Signal Process.*, vol. 51, no. 8, pp. 2177–2190, 2003.

[12] S. Azad, W. Song, and D. Tjondronegoro, "Bitrate modeling of scalable videos using quantization parameter, frame rate and spatial resolution," in *IEEE Int. Conf. on Acoust. Speech and Signal Process. (ICASSP), 2010*. IEEE, 2010, pp. 2334–2337.

[13] P. Venkat Rangan, H. Vin, and S. Ramanathan, "Designing an on-demand multimedia service," *IEEE Commun. Mag.*, vol. 30, no. 7, pp. 56–64, July 1992.

[14] J. Xiao, M. M. Hannuksela, T. Tillo, and M. Gabbouj, "A paradigm for dynamic adaptive streaming over HTTP for multi-view video," in *Advances in Multimedia Inform. Process. - PCM 2015 - 16th Pacific-Rim Conf. on Multimedia, Gwangju, South Korea, September 16-18, 2015, Proc., Part II*, 2015, pp. 410–418. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-24078-7_41

[15] T. Vu, H. T. Le, D. V. Nguyen, N. P. Ngoc, and T. C. Thang, "Future buffer based adaptation for vbr video streaming over http," in *2015 IEEE 17th International Workshop on Multimedia Signal Processing (MMSP)*, Oct 2015, pp. 1–5.

[16] Y. Zhou, Y. Duan, J. Sun, and Z. Guo, "Towards simple and smooth rate adaption for vbr video in dash," in *IEEE Visual Commun. and Image Process. Conf., 2014*, Dec 2014, pp. 9–12.

[17] H. Le, N. P. Ngoc, T. Vu, A. Pham, and T. C. Thang, "Smooth-bitrate adaptation method for http streaming in vehicular environments," in *IEEE Veh. Networking Conf. (VNC), 2014*, Dec 2014, pp. 187–188.

[18] C. Liu, I. Bouazizi, and M. Gabbouj, "Rate adaptation for adaptive http streaming," in *Proc. of the 2nd Annu. ACM Conf. on Multimedia Syst.* ACM, 2011, pp. 169–174.

[19] V. Menkovski and A. Liotta, "Intelligent control for adaptive video streaming," in *IEEE Int. Conf. on Consumer Electron. (ICCE), 2013*, Jan 2013, pp. 127–128.

[20] M. Claeys, S. Latré, J. Famaey, T. Wu, W. Van Leekwijck, and F. De Turck, "Design of a q-learning-based client quality selection algorithm for http adaptive video streaming," in *Adaptive and Learning Agents Workshop, part of AAMAS2013, Proceedings*, 2013, pp. 30–37.

[21] M. Claeys, S. Latre, J. Famaey, and F. De Turck, "Design and evaluation of a self-learning http adaptive video streaming client," *IEEE Commun. Lett.*, vol. 18, no. 4, pp. 716–719, April 2014.

[22] T. Hofeld, M. Seufert, C. Sieber, T. Zinner, and P. Tran-Gia, "Identifying qoe optimal adaptation of http adaptive streaming based on subjective studies," *Computer Networks*, vol. 81, pp. 320 – 332, 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1389128615000626

[23] R. K. P. Mok, X. Luo, E. W. W. Chan, and R. K. C. Chang, "Qdash: A qoe-aware dash system," in *Proceedings of the 3rd Multimedia Systems Conference*, ser. MMSys '12. New York, NY, USA: ACM, 2012, pp. 11–22. [Online]. Available: http://doi.acm.org/10.1145/2155555.2155558

[24] D. Jarnikov and T. Ozcelebi, "Client intelligence for adaptive streaming solutions," in *IEEE Int. Conf. on Multimedia and Expo (ICME), 2010*, July 2010, pp. 1499–1504.

[25] A. Bokani, M. Hassan, and S. Kanhere, "Http-based adaptive streaming for mobile clients using markov decision process," in *2013 20th International Packet Video Workshop*, Dec 2013, pp. 1–8.

[26] S. Garca, J. Cabrera, and N. Garca, "Quality-control algorithm for adaptive streaming services over wireless channels," *IEEE Journal of Selected Topics in Signal Processing*, vol. 9, no. 1, pp. 50–59, Feb 2015.

[27] Y. Xu, Y. Zhou, and D.-M. Chiu, "Analytical qoe models for bit-rate switching in dynamic adaptive streaming systems," *IEEE Trans. Mobile Computing*, vol. 13, no. 12, pp. 2734–2748, Dec 2014.

[28] M. Seufert, S. Egger, M. Slanina, T. Zinner, T. Hoßfeld, and P. Tran-Gia, "A survey on quality of experience of http adaptive streaming," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 1, pp. 469–492, 2015.

[29] Y. Liu, S. Dey, F. Ulupinar, M. Luby, and Y. Mao, "Deriving and validating user experience model for dash video streaming," *IEEE Transactions on Broadcasting*, vol. 61, no. 4, pp. 651–665, Dec 2015.

[30] N. Staelens, J. D. Meulenaere, M. Claeys, G. V. Wallendael, W. V. den Broeck, J. D. Cock, R. V. de Walle, P. Demeester, and F. D. Turck, "Subjective quality assessment of longer duration video sequences delivered over http adaptive streaming to tablet devices," *IEEE Transactions on Broadcasting*, vol. 60, no. 4, pp. 707–714, Dec 2014.

[31] C. Alberti, D. Renzi, C. Timmerer, C. Mueller, S. Lederer, S. Battista, and M. Mattavelli, "Automated qoe evaluation of dynamic adaptive streaming over http," in *Quality of Multimedia Experience (QoMEX), 2013 Fifth International Workshop on*. Ieee, 2013, pp. 58–63.

[32] X. Yin, V. Sekar, and B. Sinopoli, "Toward a principled framework to design dynamic adaptive streaming algorithms over http," in *Proceedings of the 13th ACM Workshop on Hot Topics in Networks*. ACM, 2014, p. 9.

[33] K. Miller, E. Quacchio, G. Gennari, and A. Wolisz, "Adaptation algorithm for adaptive streaming over http," in *19th Int. Packet Video Workshop (PV), 2012*, May 2012, pp. 173–178.

[34] T. Andelin, V. Chetty, D. Harbaugh, S. Warnick, and D. Zappala, "Quality selection for dynamic adaptive streaming over http with scalable video coding," in *Proceedings of the 3rd Multimedia Systems Conference*. ACM, 2012, pp. 149–154.

[35] M. Xing, S. Xiang, and L. Cai, "A real-time adaptive algorithm for video streaming over multiple wireless access networks," *IEEE Journal on Selected Areas in Communications*, vol. 32, no. 4, pp. 795–805, April 2014.

[36] C. D. Iskander and P. T. Mathiopoulos, "Fast simulation of diversity nakagami fading channels using finite-state markov models," *IEEE Transactions on Broadcasting*, vol. 49, no. 3, pp. 269–277, Sept 2003.

[37] C. Zhou, C.-W. Lin, and Z. Guo, "m dash: A markov decision-based rate adaptation approach for dynamic http streaming," *IEEE Transactions on Multimedia*, vol. 18, no. 4, pp. 738–751, 2016.

[38] S. Lederer, C. Müller, and C. Timmerer, "Dynamic adaptive streaming over http dataset," in *Proceedings of the 3rd Multimedia Systems Conference*, ser. MMSys '12. New York, NY, USA: ACM, 2012, pp. 89–94. [Online]. Available: http://doi.acm.org/10.1145/2155555.2155570

[39] J. Ostermann, J. Bormans, P. List, D. Marpe, M. Narroschke, F. Pereira, T. Stockhammer, and T. Wedi, "Video coding with h.264/avc: tools, performance, and complexity," *IEEE Circuits Syst. Mag.*, vol. 4, no. 1, pp. 7–28, First 2004.

Li YU received the Bachelor degree in information management and information system from the Soochow University, Suzhou, China, in 2012. Since December 2012, she has been pursuing the Ph.D. degree in electrical engineering and electronics from the University of Liverpool, Liverpool, U.K. For few months in 2014, she served as a visiting research student at the University of Electronic Science and Technology of China, Chengdu, China. For few months in 2016, she was a visiting research student at the University of Turin, Turin, Italy. Her research interests include video streaming, video coding, image and video processing.

Tammam TILLO received the Engineer Diploma in electrical engineering from the University of Damascus, Syria, in 1994, and the Ph.D. degree in electronics and communication engineering from Politecnico di Torino, Italy, in 2005. In 2004, he served as a visiting research student at Ecole Polytechnique Fdrale de Lausanne (EPFL), Lausanne, Switzerland. From 2005 to 2008, he worked as a Postdoctoral Researcher at the Image Processing Lab of Politecnico di Torino. For few months he was Invited Research Professor at the Digital Media Lab, SungKyunKwan University, Republic of Korea. In 2008 he joined Xi'an Jiaotong-Liverpool University (XJTLU), China, where he established the multimedia technology lab (MMTlab). From 2010 to 2013 he was the Head of the Department of Electrical and Electronic Engineering at XJTLU University, and was the Acting Head of the Department of Computer Science and Software Engineering from 2012 to 2013.

Jimin XIAO received the B.S. and M.E. degrees in telecommunication engineering from the Nanjing University of Posts and Telecommunications, Nanjing, China, in 2004 and 2007, respectively, and the Ph.D. degree in electrical engineering and electronics from the University of Liverpool, Liverpool, U.K., in 2013. From November 2013 to November 2014, he was a Senior Researcher with the Department of Signal Processing, Tampere University of Technology, Tampere, Finland, and an External Researcher with the Nokia Research Center, Tampere, Finland. Since December 2014, he has been a Faculty Member with Xian Jiaotong-Liverpool University, Suzhou, China. His research interests include image and video processing, computer vision, and deep learning.